CREATING AND POPULATING DATABASE

```
DROP DATABASE favorite_details;

CREATE DATABASE IF NOT EXISTS favorite_food_details;
USE favorite_food_details;

CREATE TABLE person (
    person_id SMALLINT UNSIGNED,
    fname VARCHAR(20),
    lname VARCHAR(20),
    eye_color ENUM('BR', 'BL', 'GR'),
    birth_date DATE,
    street VARCHAR(30),
    city VARCHAR(20),
    state VARCHAR(20),
    country VARCHAR(20),
    postal_code VARCHAR(20),
    CONSTRAINT pk_person PRIMARY KEY (person_id)
);



SELECT
    *
FROM
    person;
```

-- If you want to make sure that the person table does, in fact, exist, you can use the
-- describe command (or desc for short) to look at the table definition:

```
DESC person;



-- Creating favorite_food table:

CREATE TABLE favorite_food (
    person_id SMALLINT UNSIGNED,
    food VARCHAR(20),
    CONSTRAINT pk_favorite_food PRIMARY KEY (person_id , food),
    CONSTRAINT fk_fav_food_person_id FOREIGN KEY (person_id)
        REFERENCES person (person_id)
);
```

```
DESC favorite_food;


ALTER TABLE person MODIFY person_id SMALLINT UNSIGNED AUTO_INCREMENT;


DESC person;


INSERT INTO
person(person_id, fname, lname, eye_color, birth_date)
VALUES (null, 'Pradeepchandra', 'Reddy S C', 'BR', '1995-11-18');


SELECT
    person_id, fname, lname, eye_color, birth_Date
FROM
    person;


SELECT
    *
FROM
    person;
```

-- If there were more than one row in the table, however, I could add a where clause to specify that
-- I want to retrieve data only for the row having a value of 1 for the person_id column:

```
SELECT
    person_id, fname, lname, eye_color, birth_date
FROM
    person
WHERE
    person_id = 1;
```

-- While this query specifies a particular primary key value, you can use any column in
-- the table to search for rows, as shown by the following query, which finds all rows
-- with a value of 'pradepchandra' for the fname column;

```
SELECT
    *
FROM
    person
WHERE
    fname = 'pradeepchandra';
```

-- pradeepchandra has also provided information about his favorite three foods, so here
-- are three insert statements to store his food preferences

```
INSERT INTO favorite_food (person_id,food) VALUES (1,'Biriyani');
INSERT INTO favorite_food (person_id,food) VALUES (1,'Grilled Chicken');
INSERT INTO favorite_food (person_id,food) VALUES (1,'Shawarma');
INSERT INTO favorite_food (person_id,food) VALUES (1,'Fried Chicken');
```

-- Here’s a query that retrieves Pradeep's favorite foods in alphabetical order using an
-- order by clause:

```
SELECT
    *
FROM
    favorite_food
WHERE
    person_id = 1
ORDER BY food;
```

```
--

INSERT INTO person
(person_id,fname,lname,eye_color,birth_date,street,city,state,country,postal_
code)
VALUES(null, 'Susan','Smith', 'BL', '1975-11-02','23 Maple St.', 'Arlington',
'VA', 'USA', '20220');


--

SELECT
    *
FROM
    person;



--

SELECT
    *
FROM
    person
WHERE
    person_id = 2;



--

SELECT
    person_id, fname, lname, birth_date
FROM
    person;



--

UPDATE person
SET
    street = '1225 Tremont ST.',
    city = 'Boston',
    state = 'MA',
    country = 'USA',
    postal_code = '02138'
WHERE
    person_id = 1;
```

```
--

SELECT
    *
FROM
    person
WHERE
    person_id = 1;



--

SELECT
    person_id, fname, lname, state
FROM
    person;



--

SELECT
    person_id, fname, lname, state
FROM
    person
WHERE
    person_id = 1;



-- Delete

DELETE FROM person
WHERE
    person_id = 2;

SELECT * FROM person;


-- You can drop the table now since we are not going to use it again

DROP TABLE favorite_food;
DROP TABLE person;
```